

# 4 Bit Binary counter

This is a simple and basic tutorial for newbies on how to display a counting pattern on led's. It's going to be a simple binary counting on leds. Recall binary numbers (00=0, 01=1, 10=2, 11=3 and goes on). Led's will lit up in binary order.

Led's blinking pattern will be in binary form. LEDs will be connected to port-A of the STM32F0 microcontroller. I am going to make a 4-bit binary counter with an STM32F0 microcontroller and led's.

4 bit binary counter can count from 0 to 15. In binary form 0000 to 1111 range. And its total combination is 16. Let's break down this form using led on off condition. (Note when binary value 1 led turn **on** and when value 0 led turn **off**).

Decimal Value	Binary Value	Led state (ON ● OFF ●)
0	0000	● ● ● ●
1	0001	● ● ● ●
2	0010	● ● ● ●
3	0011	● ● ● ●
4	0100	● ● ● ●
5	0101	● ● ● ●
6	0110	● ● ● ●
7	0111	● ● ● ●
8	1000	● ● ● ●
9	1001	● ● ● ●
10	1010	● ● ● ●
11	1011	● ● ● ●
12	1100	● ● ● ●
13	1101	● ● ● ●
14	1110	● ● ● ●
15	1111	● ● ● ●

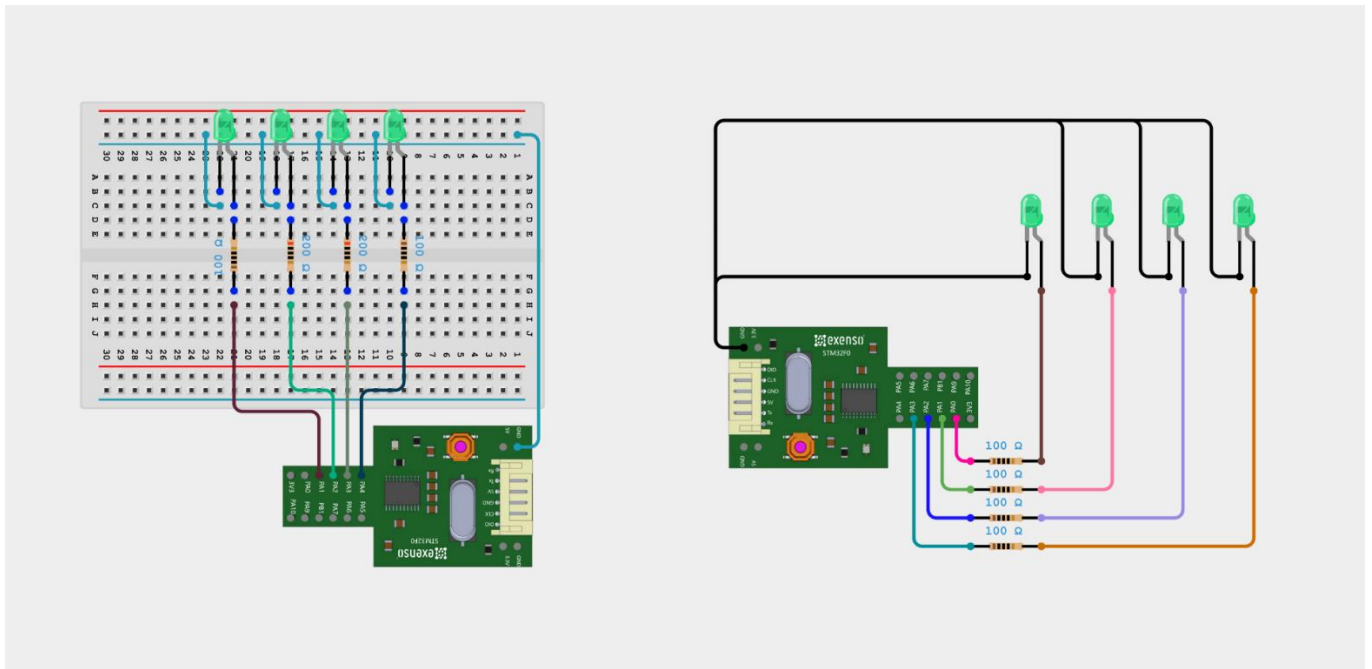
## Components Required

You will need the following components –

- 1 × Breadboard
- 1 × STM32F030F4P6
- 4 × LED
- 4 × 100Ω Resistor
- 8 × Jumper

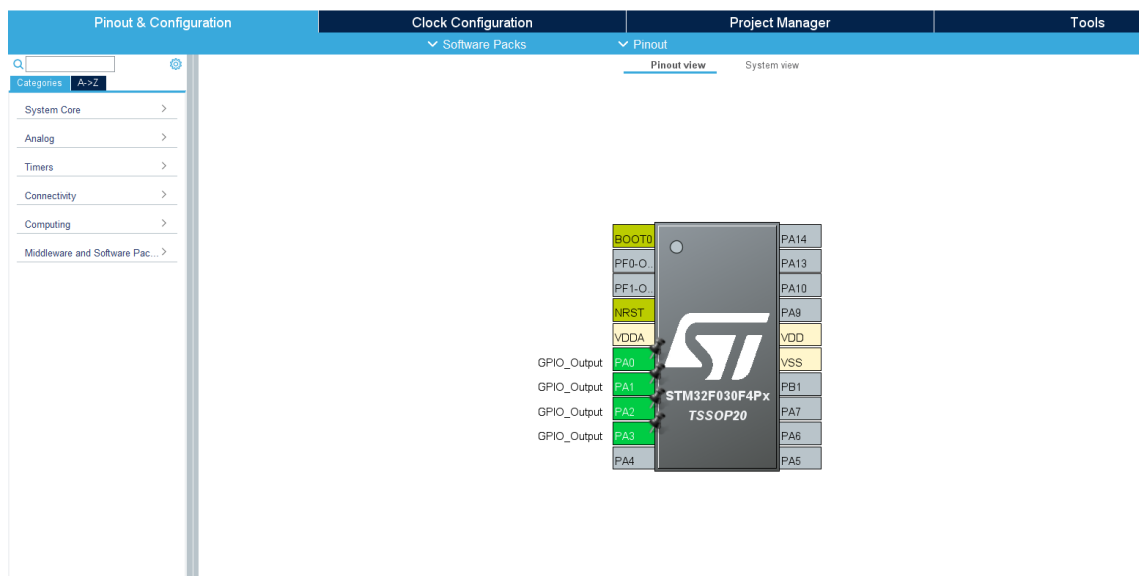
## Procedure

Follow the circuit diagram and hook up the components on the breadboard as shown in the image given below.



## STM32F0 Pin Configuration:

PA0 to PA3 set as an output.



## CODE

```

#include "main.h"

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();
    SystemClock_Config();

    MX_GPIO_Init();
    while (1)
    {
        for (uint8_t count = 0; count <= 15; count++) {
            GPIOA->ODR = count;    //ODR-Output Data Register value
            HAL_Delay(1000);    // Delay 1 second
        }
    }
}

```

```

for (uint8_t count = 0; count <= 15; count++) {
    GPIOA->ODR = count;    //ODR-Output Data Register value
    HAL_Delay(1000);    // Delay 1 second
}

```

#### For Loop:

- **uint8\_t count = 0;** : This declares a variable named count of type uint8\_t (unsigned 8-bit integer), initializing it to 0.
- **count <= 15;** : This is the loop condition. The loop will continue as long as count is less than or equal to 15.
- **count++** : This increments the value of count by 1 after each loop iteration.
- **GPIOA->ODR = count;** : This line attempts to directly write the value of the count variable to the Output Data Register (ODR) of GPIO Port A.
- **HAL\_Delay(1000);** : This line calls the HAL\_Delay function from the STM32 HAL library. It introduces a one-second (1000 milliseconds) delay between each loop iteration.